# Regular Expression Identities

*Pre-requisite knowledge: regular expressions, deterministic and non-deterministic finite automata, and regular languages.*

In this module, we examine one of the regular expression identities *(uv)\*u = u(vu)\**. We consider the following algorithm usually found on the back of a shampoo bottle for washing hair:

1. Wet hair. (W)
2. Add shampoo to hair. (S)
3. Lather hair. (L)
4. Rinse hair. (R)
5. Repeat at Step 2.

We could form a regular expression for this process as W(SLR)*SLR. First, one must wet their hair. This is followed by at least one cycle of shampoo-lather-rinse. Applying the regular expression identity, *(uv)\*u = u(vu)\**, this regular expression may be re-written as WSL(RSL)*R.
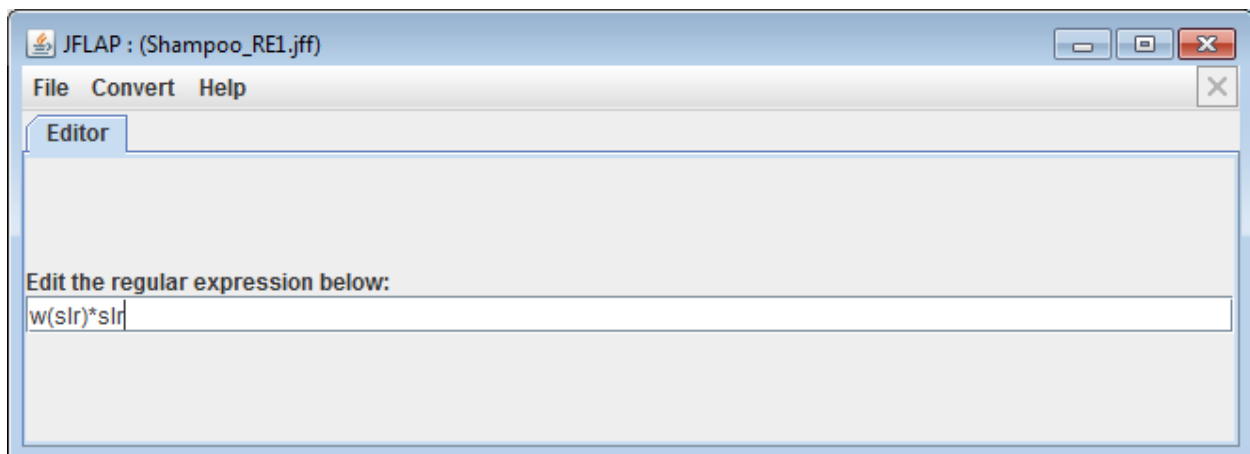
Next, we will use JFLAP to show that the two regular expressions are equivalent:

1. W(SLR)*SLR
2. WSL(RSL)*R

To do so, we will create each regular expression separately and convert each to an NFA, then to a DFA. Once both DFAs are created, we can then compare the DFAs and check for equivalence.
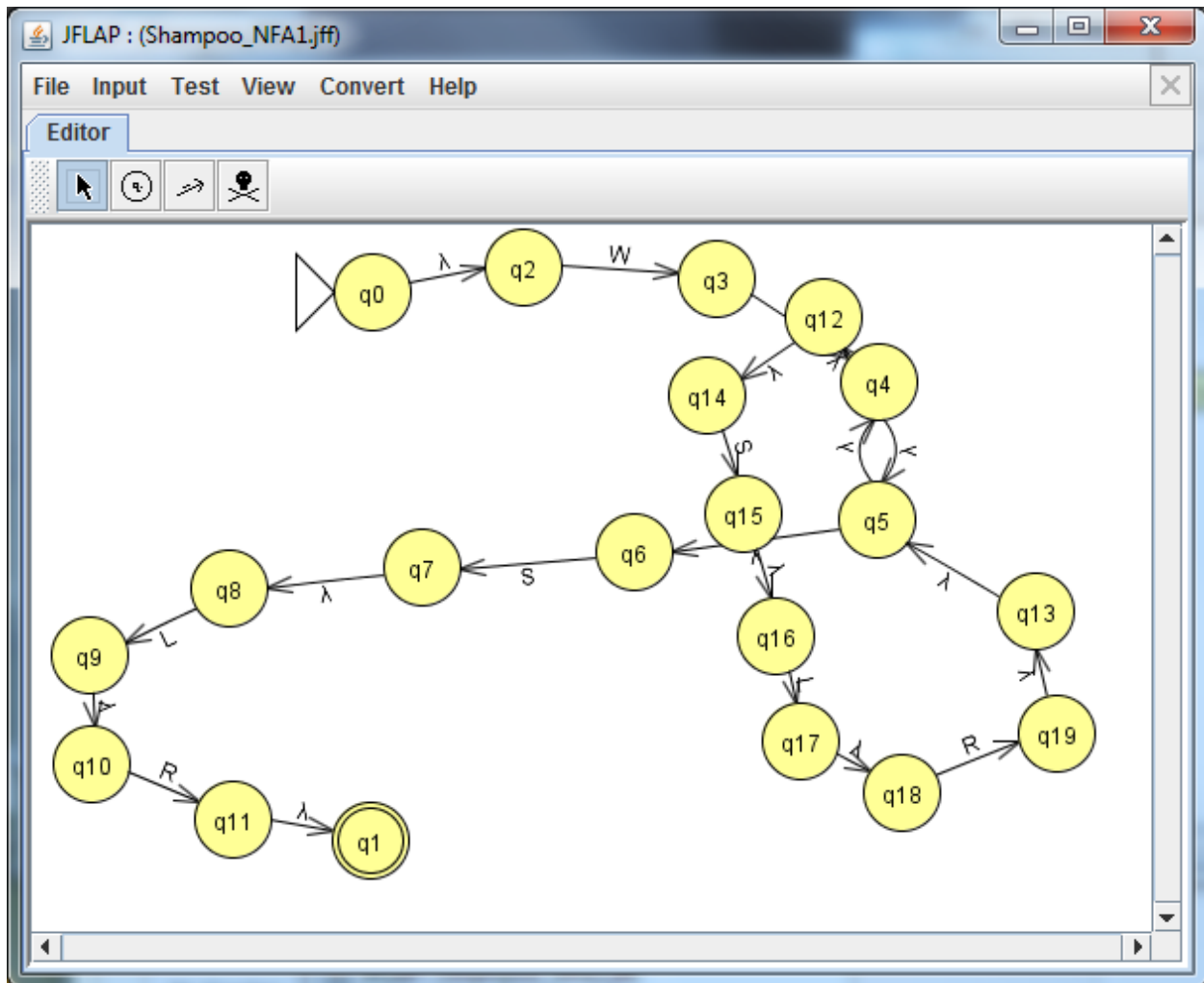
**Regular Expression 1, W(SLR)*SLR**

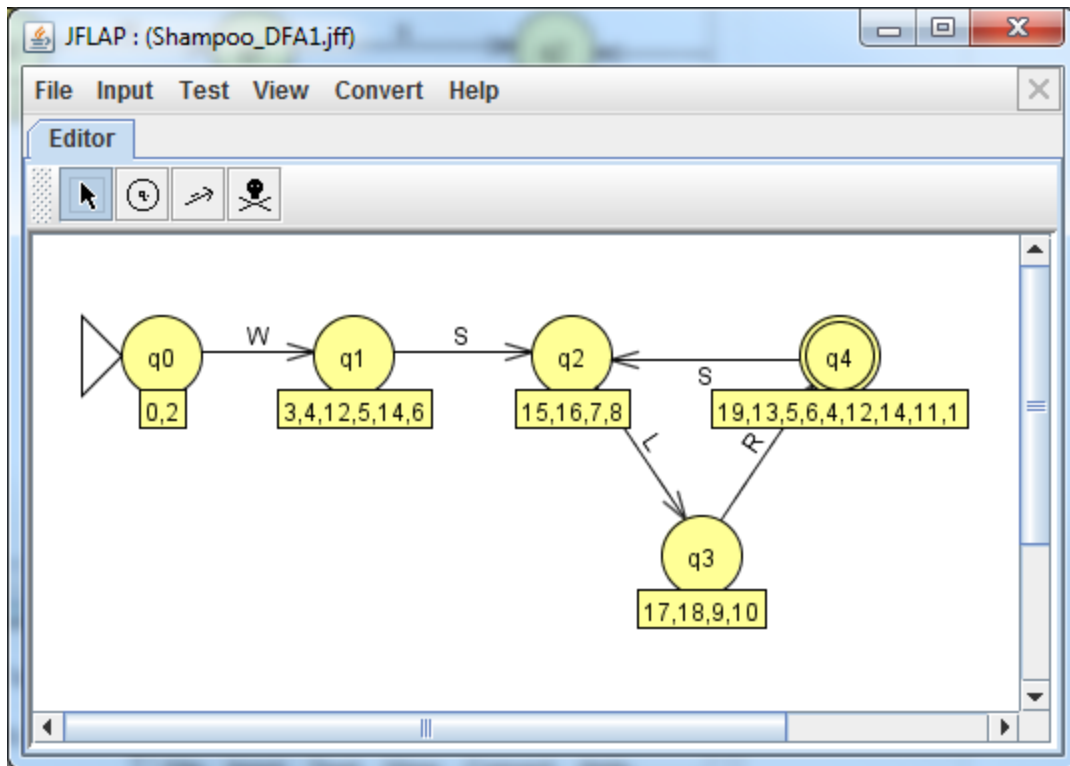Enter the first regular expression, W(SLR)*SLR, into JFLAP.



Convert this to a DFA by selecting Convert > RE to FA. On the next window, select Step to Completion and Export the FA.

**Try It!** If you are not familiar with converting regular expression to FA, review that module. Go ahead and finish the conversion from regular expression to FA. Export this to its own work area. Change the layout as needed.

The next step is to convert this NFA to a DFA. Again, if you are not familiar with this process, review the module on converting NFA to DFA. Change the diagram layout to make it more readable. You should have the DFA below.

**Regular Expression 2, WSL(RSL)*R**

**Try It!** Enter the expression WSL(RSL)*R into JFLAP as a regular expression. Convert this to a DFA as above via an NFA. This results in the DFA below.
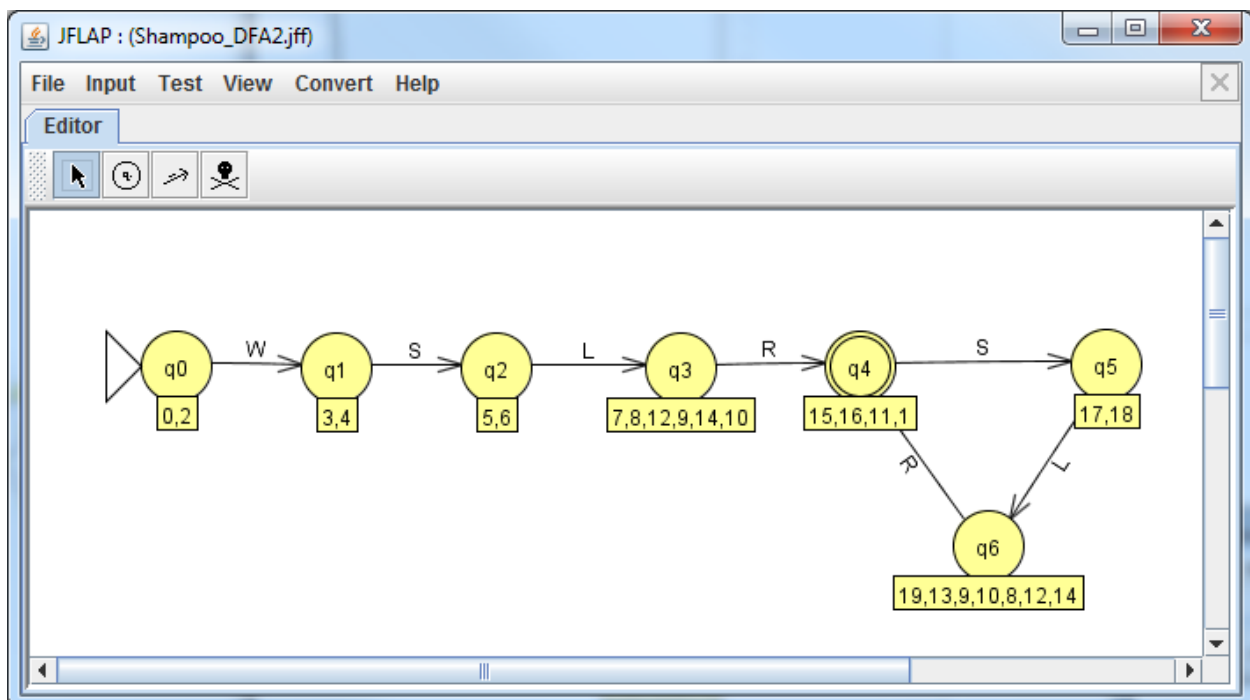
Lastly, we check the equivalence of the two DFAs.

**Try It!** With the two DFA's open in JFLAP, verify that the two automata are equivalent.

---

Reference:

Peter Linz, "An Introduction to Formal Languages and Automata" 5th edition, Jones and Bartlett, 2011.